06 | 全链路压测：系统整体容量保障的“核武器”（下）

2021-05-24 吴骏龙

《容量保障核心技术与实战》 课程介绍 >

讲述：吴骏龙
时长 10:51 大小 9.95M

你好，我是吴骏龙。

上一讲，我为你讲解了在正式实施全链路压测前，我们要做的三项改造工作，包括数据隔离、中间件改造和应用服务改造。这一讲，我们就正式进入两项压测工作：压测模型构建和压测流量构造，把全链路压测的建设过程完整展示给你。除了技术工作之外，在这一讲中我还会与你分享全链路压测的组织协调和运营工作，它们对全链路压测的完整落地同样起到至关重要的作用。

我们先看看如何构建压测模型。

两项压测工作：压测模型构建

构建压测模型的重点是准确度，如果模型与真实场景相差过大，那么压测结果的可参考性将会大打折扣，下面是一些典型的由于压测模型不准确导致压测结果无效的反面教材：

下单链路中，压测用户没有使用红包，导致对营销服务的压测结果偏优。

压测用户数据未考虑 sharding 分布，导致数据库单片过热。

压测用户数量过少，使用有限的压测用户反复下单后，导致单个用户订单量过多。

压测商户数量过少，压测时针对单个商户的操作过于密集，导致菜品扣减库存的锁争抢激烈。

压测模型包含业务模型和数据两部分，我再来通过几个实例讲解一下如何构建尽可能真实的场景。

实例一: 读请求

读请求由于不会对数据造成污染，因此可以直接使用真实请求和数据进行回放。

压测场景：商家列表及关键词查询。

业务模型：拉取线上日志，根据真实接口比例关系进行回放。

数据：拉取线上日志，使用真实数据。

实例二: 写请求

写请求一般需要单独构造压测模型，并做好数据隔离和清理工作。

压测场景：用户下单

业务模型：根据生产监控或日志，获取下单场景的链路信息，观察接口调用情况和上下游依赖，当然，你也可以写一个系统帮你做这个事。产品、研发和测试共同评审链路的完整性。另外，评估业务改造点，比如需要对支付和短信等环节进行 Mock。

数据：构建测试用户、测试商户、测试菜品等数据，数量上与线上真实情况等比例缩放；及时对压测数据进行清理，或使用影子表。

归纳一下，压测模型构建的核心要点是，要利用好生产环境的各种信息来帮助我们**构建贴近真实业务的压测模型**。生产环境是个聚宝盆，请求的依赖关系、调用比例、数据特征都是我们构建压测模型的素材，将这些数据抽取出来再进行精加工，即可得到贴合实际的压测模型。

两项压测工作：压测流量构造

有了压测模型和数据，最后临门一脚就是构造压测流量进行施压。全链路压测对于压测流量构造的技术选型主要取决于流量的规模，如果规模不大，传统的压测工具是可以支持的，如 JMeter、Locust、nGrinder 等；如果是大规模流量乃至超大规模流量（百万请求量级），成本就会比较高。对于后者，可以考虑自研一套压测平台，这也是很多大厂的做法，我在下一讲会专门展开这部分内容，敬请关注。

我们来总结一下，全链路压测的建设过程可以归纳为两个重点：首先，通过中间件改造和应用服务改造，保证压测流量的完整性和可识别性，并保证压测数据与真实数据隔离开；其次，利用生产环境的各类信息，构建贴近真实场景的压测模型，并通过构造大规模压测流量实施全链路压测。

这些工作都完成后，全链路压测在技术层面的建设就基本告一段落了。

全链路压测的组织协调和运营工作

说了那么多，也许你会觉得建设全链路压测的技术难度还是挺高的，但我想告诉你的是，除了技术工作，组织协调和运营工作其实更难。这就好比新冠肺炎疫情的防控，全世界都知道中国的成功经验，但有几个国家能成功复制中国的防疫举措呢？

关于全链路压测建设时会涉及到的组织协调工作，通过全链路压测的建设过程相信你也看到了，其中光中间件改造和业务改造两项工作，就几乎覆盖了大半个技术团队，要同时协调那么多团队的工作安排，难度不小吧？

我认为，推动全链路压测这样的“航空母舰”项目，是需要自上而下的，但不一定非要强推。我在阿里本地生活工作时，技术团队建立了 **“Program 机制”** ，这是一种针对跨团队大型项目的推动机制，由 CTO 直接牵头和授权，对公司内部需要推动的技术改造类项目，进行必要性和优先级评定。

从项目跟进的角度来说，所有公共团队（如基础设施团队、大数据团队等）和业务团队的技术 Leader 需要定期参加会议，在会议上对这些项目的进展和风险进行讨论，业务团队必须在约定时间内完成公共团队的技术改造需求，而公共团队则需要提供合理的方案，并提供足够的支持。

Program 机制为基础设施团队推动技术改造类项目提供了一个强有力的抓手，**动态平衡了业务实现与技术改造之间的关系**，使得业务团队必须腾出一部分时间进行技术升级，而不是埋头沉迷于业务迭代。

全链路压测就是众多技术改造类项目的一员。在我目前所在的这家创业公司，同样是依托于类似的机制，仅仅 2 个多月的时间，便从 0 到 1 推动完成了全链路压测的核心改造工作，所以你也大可不必把全链路压测想的很难，可以尝试用类似的思路去推动它。

组织协调很难，另一个更有难度的问题是，在全链路压测建设完成后，如何将其有效地运营起来，明确每个参与团队要做什么事，做这些事的规范是什么，做的不好的后果是什么等等，这样才能将全链路压测的价值最大化地固化下来。

我的经验是，全链路压测是需要有一个集中式的团队去管理的，这个团队不需要很多人，但是需要被充分授权。可能你会问，光授权也没用啊，别人不听你的怎么办？**这时候就需要通过一些规范去约束和管控了**。

我在做全链路压测运营工作时，建立了两项规范，首先是**全链路压测的常态化执行制度**，每周三晚间低峰期执行全链路压测，核心链路的技术人员和运维人员必须现场值守，其余技术人员可以远程值守。值守人员需要严密关注业务指标，如果出现服务可用性问题或资损问题，及时报告压测团队暂停压测。

如果压测过程中出现服务瓶颈，我们有时候会执行一些降级操作以观察效果，这时候值守人员也应配合操作，如果因为未有效值守导致线上问题，需要承担连带责任。

此外，全链路压测能够暴露出整体系统的容量隐患，但仅仅将问题暴露出来还是不够的，我们需要确认这些问题得到重视和解决，才是真真正正地消除了容量风险，我建立的第二项规范，就是用来驱动全链路压测时所发现问题的及时改进，称之为**容量问题分级规范**。

这项规范根据容量风险的严重程度划定了不同的等级，每个等级对应不同的解决时限要求，越严重的风险，越是需要快速解决，或至少有临时措施。我们会定期统计问题解决的时长达标率，以此作为所有技术团队绩效考评的一个参考标准。

严重级别	S0（严重风险）	S1（高风险）	S2（中高风险）	S3（低风险）
需解决时间	问题暴露后次日高峰期前	问题暴露后24小时内	问题暴露后72小时内	问题暴露后1周内
维度（压测峰期间）	超过半数的服务器/实例CPU利用率达到或超过100%	超过半数的服务器/实例CPU利用率达到或超过90%	超过半数的服务器/实例CPU利用率达到或超过80%	少量（<50%）服务器/实例CPU利用率达到或超过80%
	超过半数的服务器/实例内存使用率达到100%	超过半数的服务器/实例内存使用率达到90%	超过半数的服务器/实例内存使用率达到80%	少量（<50%）服务器/实例内存使用率达到或超过80%
	接口响应时间增加200%以上（相比高峰期）	接口响应时间增加100%以上（相比高峰期）	接口响应时间增加100%以下（相比高峰期），但对业务有感	接口响应时间增加100%以下（相比高峰期），但对业务轻微有感
	持续出现超过1k/s的异常	持续出现超过500/s的异常	波段出现超过1k/s的异常	持续出现少于500/s的异常
	带宽打满（占用100%带宽）	带宽接近打满（占用超过80%带宽）	带宽存在隐患（占用超过70%带宽）	带宽存在其他轻微风险
	其他性能指标出现严重瓶颈或打满，并可能影响业务	其他性能指标达到满负荷的90%，并可能影响业务	其他性能指标达到满负荷的80%，并可能影响业务	其他性能指标存在一定隐患，对业务影响轻微
	服务问题导致压测量无法达到目标值，差值≥0.5倍	服务问题导致压测量无法达到目标值，差值<0.5倍	服务问题导致压测量虽然达到目标值，但存在大量抖动	服务问题导致压测量虽然达到目标值，但存在少量抖动
	其他可能严重影响关键路径服务的隐患	其他可能部分影响关键路径服务的隐患	其他可能严重影响非关键路径服务的隐患	其他基本对业务无影响，但由压测发现的问题

极客时间

总结一下，推动全链路压测的落地，不仅仅是一项技术工作，组织协调和运营工作同样重要，否则还是很容易失败的。我比较倡导通过建立机制和流程规范的方式，自上而下去联动和管理多个团队之间的工作，定好的规范要及时跟进并督促执行，尽早暴露风险。

总结

关于全链路压测的内容比较多，我们来好好总结一下。全链路压测通过模拟未来流量峰值提前发生，将不确定问题转化为确定性问题，从而达到提前暴露系统整体容量问题的目的。

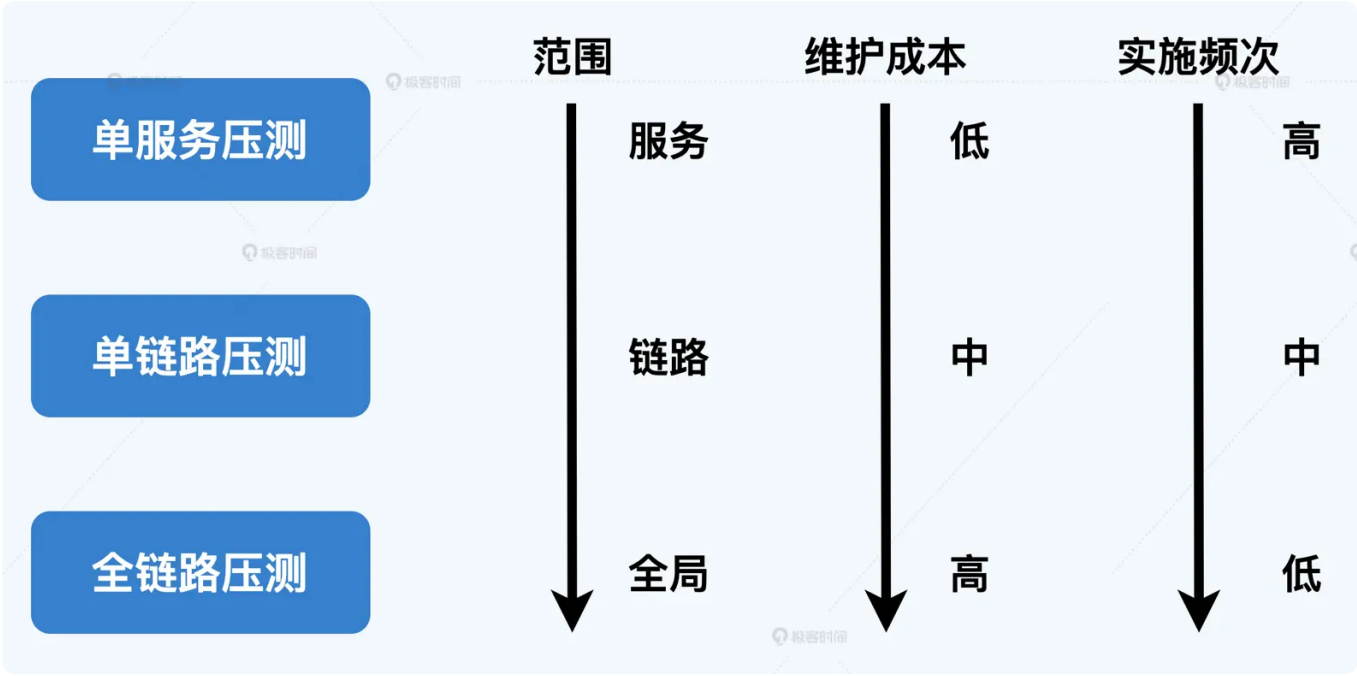
全链路压测的建设过程，涉及到数据隔离、中间件改造、业务服务改造、压测模型构建和压测流量构造这五项工作，有一定的技术难度和改造量，虽然我在讲解中提供了多种方案，但你在制定技术方案时还是需要平衡好投入产出比。

例如，公司大量采用开源技术作为基础设施，业务场景也比较简单，这时候完全可以不用去动这些基础设施，可以直接在业务层进行压测数据的逻辑隔离。构造流量也可以直接使用成熟的开源工具（JMeter、Locust 等）。一句话，适合的才是最好的。

全链路压测不是单一的技术问题，组织协调和运营工作也需要重点考虑，建立一支强有力的全链路压测团队，通过流程和机制的制定，去管理和规范各个团队的工作，是我给到的经验之谈。Program 机制、全链路压测常态化执行制度和容量问题分级规范，是我给出的三项具体可操作的方法，也是我推动或利用的比较成功的例子，希望能够给你带来一些启发。

最后，河有两岸，事有两面，全链路压测也不是银弹，无法解决所有问题，将所有容量问题全部交给全链路压测兜底，不再做单链路压测或单服务压测，是错误的实践。全链路压测的实施成本较高，因此其实施频率一般是远远低于业务变更频率的。

全链路压测的擅长点是定期摸底系统整体容量，而常态的容量保障工作应当覆盖每个业务各个接口，这些毛细血管依然需要单链路压测和单服务压测去保障。



课后讨论

假设我们通过全链路压测得到结论，系统整体能够承载 1000TPS（每秒下 1000 单）的负载，但实际业务达到这个负载时，系统却出现了不稳定的情况，你觉得在全链路压测工作中可能有哪些地方我们考虑的不够周全，从而导致了这一问题？欢迎在评论区与我交流你的想法。

分享给需要的人，Ta订阅后你可得 16 元现金奖励

赞 0 提建议

© 版权归极客邦科技所有，未经许可不得传播售卖。 页面已增加防盗追踪，如有侵权极客邦将依法追究其法律责任。

上一篇 05 | 全链路压测：系统整体容量保障的“核武器”（上）

下一篇 07 | 工具进化：如何实现一个分布式压测平台

精选留言 (3)

写留言

莫_努力增肥25斤
2021-05-25

测试场景也是一方面，一般都是通过日志找出高频接口的调用比例，但低频接口也会出问题，而且往往从来未压测和优化过，甚至没几个人知道这功能存在，由于太低频在各种监控和统计里完全透明，一出问题就是阻塞db。

作者回复: 说的非常好！低调用量≠低风险

1

dalek
2021-06-04

针对于常态化压测中值守人员的问题，是否可以使用无人值守或者减少值守人员的方式来做？比如采集相关指标对压测配置修改、压测状态同步...这个有尝试过么？是否有效？

展开

作者回复: 这是一个很好的思考方向，狭义的压测执行期间的无人值守，在技术上的难度并不大，由压测平台按照预置的策略去自动化施压，并对接外部监控系统（指标需要提前设置），在识别到风险时主动熔断压测或变更策略，完全可以做到不需要人的干预。风险在于对接的监控指标的完备性，但一般互联网公司的NOC团队都是随时处于值班状态的，可以兜底风险。

广义的说，我们还希望能做到压测全流程的无人值守，包括压测前（准备压测脚本&数据）和压测后（分析结果&输出风险项）的低人力甚至无人力投入，这就有很大的难度了。业内是有一些实践的，比如通过埋点的方式自动从流量入口梳理链路；包括类似于你提到的通过定期采集线上各接口的流量数据，去反向对齐压测时各接口的压力配比，等等，都是在朝着这个方向努力。

Roy Liang
2021-05-24

原因可能是全链路压测不能完全复刻外部依赖接口，例如银联支付等场景

作者回复: 回答的不错！对外部依赖的第三方接口调用处理不好，很容易成为全链路压测场景失真的一个因素。改进这一点的策略就是尽可能仿真，比如针对支付场景，我们可以mock支付回调，并按照真实回调的响应时间设置一定的延时，甚至可以制造一些波动，来尽可能逼近实际情况。